AKADEMIA GÓRNICZO-HUTNICZA
IM. STANISŁAWA STASZICA
W KRAKOWIE

Metody Obliczeniowe w Nauce i Technice

Sprawozdanie z laboratorium 6 – Rozwiązywanie równań i układów równań nieliniowych

Michał Szafarczyk

gr. Śr. 17:50 – 19:20

Narzędzia i sprzęt wykorzystany do zrealizowania ćwiczenia

Komputer z systemem Windows 10 x64 Home

Procesor: Intel Core i7-10750H @2.60 GHz / 5.00 GHz

Pamięć RAM: 32 GB

Język: Python 3.9

Środowisko: PyCharm

Użyte biblioteki pythonowskie:

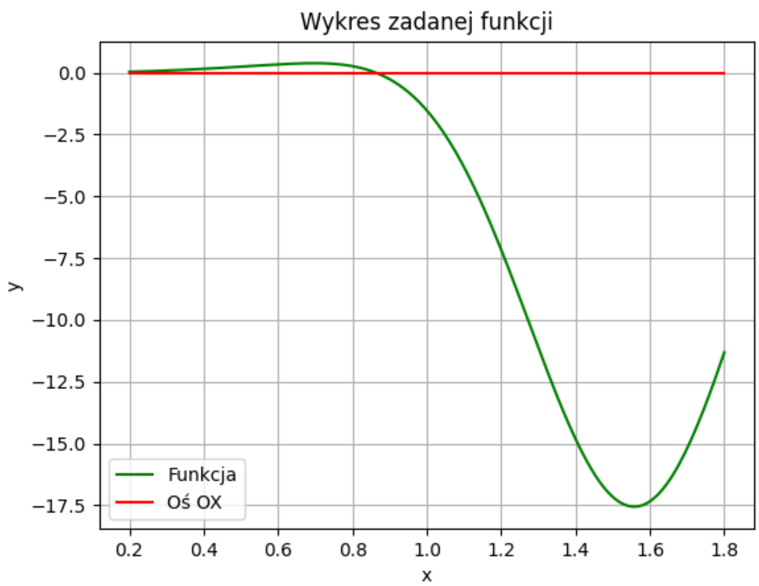
- Numpy – do wykonywania różnych operacji na liczbach
- Matplotlib – dla rysowania wykresów
- Sympy – blibioteka do wykonywania operacji na równaniach z wieloma niewiadomymi

1. Zadana funkcja:

Przekazana wraz z zadaniem funkcja, dla której będziemy wyznaczać miejsca zerowe, zadana jest na przedziale $[0.2, 1.8]$ i dana wzorem:

$$f(x) = x^2 - m \cdot sin(x)^n$$

$$gdzie \quad m = 20, \qquad n = 12$$



Wykres 1.1 – Zadana funkcja

2. Metoda Newtona:

Dla użycia metody Newtona potrzebujemy znać pierwszą pochodną zadanej funkcji; w tym celu ją obliczymy ręcznie:

$$f'(x) = 2 \cdot x - m(n) \cdot sin(x)^{n-1} \cdot cos(x)$$

$$gdzie \quad m = 20, \qquad n = 12$$

Metoda Newtona używa wzoru rekurencyjnego, który wykorzystuje pochodną zadanej funkcji oraz nią samą dla obliczania kolejnych przybliżeń miejsca zerowego:

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$$

Działanie algorytmu rozpoczynamy od zadania pewnego punktu startowego (znajdującego się w przedziale $[a, b]$.

3. Metoda siecznych:

Metoda siecznych jest zbliżona do metody Newtona. Pierwszą różnicą jest liczba punktów startowych – dla metody siecznych potrzebujemy 2 punktów dla rozpoczęcia działania. Metoda przy każdym wywołaniu rekurencyjnym prowadzi sieczną przez 2 ostatnie punkty przybliżenia. Wzór rekurencyjny przedstawia się wzorem:

$$x_{k+2} = x_{k+1} - \frac{x_{k+1} - x_k}{f(x_{k+1}) - f(x_k)} f(x_{k+1})$$

Wzór ten jest tym samym wzorem, którego używamy w metodzie Newtona, jednak zamiast faktycznej wartości $f'(x)$ używamy jej przybliżenia za pomocą ilorazu różnicowego

$$f'(x) \approx \frac{x_{k+1} - x_k}{f(x_{k+1}) - f(x_k)}$$

4. Warunki końca:

Dla obu przedstawionych metod, jako, że są to metody wyznaczania przybliżonego rozwiązania, potrzebujemy zdefiniować, w którym momencie wynik będzie na tyle zadowalający, że możemy zakończyć obliczanie kolejnych przybliżeń. Na początku zadajemy pewną liczbę ρ, która określać będzie z jaką dokładnością chcemy wyznaczyć rozwiązanie.

Test wartości:

Przedstawia się wzorem:

$$|f(x_k)| < \rho$$

Test sprawdza czy wartość dla obecnie wyznaczonego rozwiązania jest bliska 0. Im ρ mniejsze, tym rozwiązanie będzie dawało wynik bliższy 0.

Test długości kroku:

Przedstawia się wzorem:

$$|x_{k+1} - x_k| < \rho$$

Test sprawdza jak daleko przemieściło się rozwiązanie przy ostatniej iteracji. Jeżeli kroki stają się bardzo niewielkie, możemy stwierdzić, że rozwiązanie jest na tyle blisko faktycznego rozwiązania, że jest wystarczające.

Test liczby iteracji:

Wprowadzamy je z 2 powodów: ze względu na to, że nie dla każdego punktu startowego otrzymamy rozwiązanie oraz ponieważ możemy chcieć mieć możliwość zadać skończoną ilość iteracji dla wyznaczenia rozwiązania. Przy każdym teście, który będziemy wykonywali, zadamy pewną dużą liczbę n, która będzie służyła, jako awaryjny warunek końca w przypadku niepowodzenia znalezienia miejsca zerowego (które się nie powiedzie przykładowo przez wybranie złośliwego punktu startowego).

Wartość ρ:

Dla testów przyjmiemy wartości $\rho \in [10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}, 10^{-7}, 10^{-9}]$.

5. Testy:

Sigma:	10^-1			10^-2			10^-3	
Start	Iteracje	Wynik	Start	Iteracje	Wynik	Start	Iteracje	Wynik
0.3	0	0.3	0.3	2	0.074964	0.3	4	0.018741
0.4	1	0.198566	0.4	2	0.099282	0.4	4	0.024821
0.5	1	0.235837	0.5	3	0.058956	0.5	4	0.029478
0.6	1	0.192345	0.6	2	0.096172	0.6	4	0.024043
0.7	14	2.066397	0.7	15	2.067173	0.7	15	2.067173
0.8	2	0.868805	0.8	3	0.863887	0.8	3	0.863887
0.9	1	0.869223	0.9	2	0.863909	0.9	2	0.863909
1	2	0.873085	1	3	0.864183	1	4	0.863762
1.1	3	0.86806	1.1	4	0.863853	1.1	4	0.863853
1.2	3	0.874945	1.2	4	0.864361	1.2	5	0.863763
1.3	3	0.877372	1.3	4	0.864637	1.3	5	0.863765
1.4	3	0.865262	1.4	3	0.865262	1.4	4	0.863772
1.5	1	0.230515	1.5	3	0.057627	1.5	4	0.028813
1.6	9	0.871531	1.6	10	0.864057	1.6	11	0.863761
1.7	3	2.065404	1.7	4	2.067169	1.7	4	2.067169

Sigma:	10^-4			10^-7			10^-9	
Start	Iteracje	Wynik	Start	Iteracje	Wynik	Start	Iteracje	Wynik
0.3	5	0.009371	0.3	10	0.000293	0.3	14	1.83E-05
0.4	6	0.006205	0.4	11	0.000194	0.4	14	2.42E-05
0.5	6	0.00737	0.5	11	0.00023	0.5	14	2.88E-05
0.6	6	0.006011	0.6	11	0.000188	0.6	14	2.35E-05
0.7	15	2.067173	0.7	16	2.067174	0.7	16	2.067174
0.8	4	0.863761	0.8	5	0.863761	0.8	5	0.863761
0.9	3	0.863761	0.9	4	0.863761	0.9	4	0.863761
1	4	0.863762	1	5	0.863761	1	5	0.863761
1.1	5	0.863761	1.1	6	0.863761	1.1	6	0.863761
1.2	5	0.863763	1.2	6	0.863761	1.2	6	0.863761
1.3	5	0.863765	1.3	6	0.863761	1.3	6	0.863761
1.4	4	0.863772	1.4	5	0.863761	1.4	6	0.863761
1.5	6	0.007203	1.5	11	0.000225	1.5	14	2.81E-05
1.6	11	0.863761	1.6	12	0.863761	1.6	12	0.863761
1.7	5	2.067174	1.7	5	2.067174	1.7	6	2.067174

Tabela 5.1 – Testy dla metody Newtona | Test wartości

Sigma:	10^-1			10^-2			10^-3	
Start	Iteracje	Wynik	Start	Iteracje	Wynik	Start	Iteracje	Wynik
0.3	5000	2.9E-163	0.3	5000	2.9E-163	0.3	5000	2.9E-163
0.4	5000	2.4E-163	0.4	5000	2.4E-163	0.4	5000	2.4E-163
0.5	5000	7E-163	0.5	5000	7E-163	0.5	5000	7E-163
0.6	5000	8.4E-163	0.6	5000	8.4E-163	0.6	5000	8.4E-163
0.7	12	1.837737	0.7	5000	2.067174	0.7	5000	2.067174
0.8	5000	0.863761	0.8	5000	0.863761	0.8	5000	0.863761
0.9	5000	0.863761	0.9	5000	0.863761	0.9	5000	0.863761
1	5000	0.863761	1	5000	0.863761	1	5000	0.863761
1.1	5000	0.863761	1.1	5000	0.863761	1.1	5000	0.863761
1.2	5000	0.863761	1.2	5000	0.863761	1.2	5000	0.863761
1.3	5000	0.863761	1.3	5000	0.863761	1.3	5000	0.863761
1.4	5000	0.863761	1.4	5000	0.863761	1.4	5000	0.863761
1.5	5000	5E-163	1.5	5000	5E-163	1.5	5000	5E-163
1.6	5000	0.863761	1.6	5000	0.863761	1.6	5000	0.863761
1.7	0	1.7	1.7	5000	2.067174	1.7	5000	2.067174

Sigma:	10^-4			10^-7			10^-9	
Start	Iteracje	Wynik	Start	Iteracje	Wynik	Start	Iteracje	Wynik
0.3	5000	2.9E-163	0.3	5000	2.9E-163	0.3	5000	2.9E-163
0.4	5000	2.4E-163	0.4	5000	2.4E-163	0.4	5000	2.4E-163
0.5	5000	7E-163	0.5	5000	7E-163	0.5	5000	7E-163
0.6	5000	8.4E-163	0.6	5000	8.4E-163	0.6	5000	8.4E-163
0.7	5000	2.067174	0.7	5000	2.067174	0.7	5000	2.067174
0.8	5000	0.863761	0.8	5000	0.863761	0.8	5000	0.863761
0.9	5000	0.863761	0.9	5000	0.863761	0.9	5000	0.863761
1	5000	0.863761	1	5000	0.863761	1	5000	0.863761
1.1	5000	0.863761	1.1	5000	0.863761	1.1	5000	0.863761
1.2	5000	0.863761	1.2	5000	0.863761	1.2	5000	0.863761
1.3	5000	0.863761	1.3	5000	0.863761	1.3	5000	0.863761
1.4	5000	0.863761	1.4	5000	0.863761	1.4	5000	0.863761
1.5	5000	5E-163	1.5	5000	5E-163	1.5	5000	5E-163
1.6	5000	0.863761	1.6	5000	0.863761	1.6	5000	0.863761
1.7	5000	2.067174	1.7	5000	2.067174	1.7	5000	2.067174

Tabela 5.2 – Testy dla metody Newtona | Test długości kroku

Sigma:	10^-1			10^-2			10^-3	
Start	Iteracje	Wynik	Start	Iteracje	Wynik	Start	Iteracje	Wynik
0.3	1	0.163181	0.3	5000	0.074894	0.3	5000	0.074894
0.4	1	0.180569	0.4	5000	0.099036	0.4	5000	0.099036
0.5	1	0.191454	0.5	5000	0.12253	0.5	5000	0.12253
0.6	1	0.196654	0.6	5000	0.137704	0.6	5000	0.137704
0.7	1	0.191993	0.7	5000	0.123958	0.7	5000	0.123958
0.8	1	0.127732	0.8	51	0.062299	0.8	5000	0.048323
0.9	2	0.250835	0.9	5000	0.178566	0.9	5000	0.178566
1	1	0.244202	1	42	0.866631	1	1974	0.863583
1.1	1	0.242932	1.1	5000	0.86811	1.1	5000	0.86811
1.2	1	0.241941	1.2	2	0.250077	1.2	24	0.863553
1.3	1	0.241435	1.3	2	0.247181	1.3	28	0.863228
1.4	1	0.241198	1.4	2	0.245897	1.4	480	0.862721
1.5	1	0.241126	1.5	2	0.245522	1.5	5000	0.889225
1.6	1	0.241195	1.6	2	0.245881	1.6	714	0.862697
1.7	1	0.241434	1.7	2	0.247171	1.7	27	0.863322

Sigma:	10^-4			10^-7			10^-9	
Start	Iteracje	Wynik	Start	Iteracje	Wynik	Start	Iteracje	Wynik
0.3	57	nan	0.3	5000	0.074894	0.3	5000	0.074894
0.4	5000	0.20670716549791085	0.4	5000	0.099036	0.4	5000	0.099036
0.5	5000	0.19319598716576758	0.5	5000	0.12253	0.5	5000	0.12253
0.6	5000	0.190815143813774	0.6	5000	0.137704	0.6	5000	0.137704
0.7	5000	0.16357298048141028	0.7	5000	0.123958	0.7	5000	0.123958
0.8	5000	0.07014835119096643	0.8	5000	0.048323	0.8	5000	0.048323
0.9	5000	0.14007232299946495	0.9	5000	0.178566	0.9	5000	0.178566
1	18	0.8637131822362345	1	5000	0.749137	1	5000	0.749137
1.1	5000	0.8662812286622921	1.1	5000	0.86811	1.1	5000	0.86811
1.2	64	0.8637355449680277	1.2	82	0.863761	1.2	110	0.863761
1.3	5000	0.8730608525445581	1.3	5000	0.830149	1.3	5000	0.830149
1.4	4750	0.8638200004409962	1.4	5000	0.874509	1.4	5000	0.874509
1.5	5000	0.8390365399229451	1.5	5000	0.889225	1.5	5000	0.889225
1.6	831	0.8637625015020371	1.6	5000	0.878988	1.6	5000	0.878988
1.7	5000	0.8355452194401756	1.7	5000	0.859899	1.7	5000	0.859899

Tabela 5.3 – Testy dla metody siecznych | Test długości kroku | Start_2 = 0.2

Sigma:	10^-1			10^-2			10^-3	
Start	**Iteracje**	**Wynik**	**Start**	**Iteracje**	**Wynik**	**Start**	**Iteracje**	**Wynik**
0.3	1	0.119986	0.3	6	0.098784	0.3	5000	0.074894
0.4	1	0.133198	0.4	52	0.099894	0.4	5000	0.099036
0.5	1	0.142043	0.5	5000	0.12253	0.5	5000	0.12253
0.6	1	0.146488	0.6	5000	0.137704	0.6	5000	0.137704
0.7	1	0.142497	0.7	5000	0.123958	0.7	5000	0.123958
0.8	1	0.095011	0.8	1	0.095011	0.8	5000	0.048323
0.9	1	0.294134	0.9	5000	0.178566	0.9	5000	0.178566
1	1	0.220505	1	41	0.862335	1	1973	0.863848
1.1	1	0.209349	1.1	5000	0.86811	1.1	5000	0.86811
1.2	1	0.20556	1.2	12	0.862129	1.2	26	0.863908
1.3	1	0.203942	1.3	26	0.864052	1.3	184	0.863728
1.4	1	0.203231	1.4	110	0.865024	1.4	2040	0.863798
1.5	1	0.203024	1.5	5000	0.889225	1.5	5000	0.889225
1.6	1	0.203222	1.6	242	0.864419	1.6	1864	0.863911
1.7	1	0.203937	1.7	26	0.863862	1.7	26	0.863862

Sigma:	10^-4			10^-7			10^-9	
Start	**Iteracje**	**Wynik**	**Start**	**Iteracje**	**Wynik**	**Start**	**Iteracje**	**Wynik**
0.3	57	nan	0.3	5000	0.074894	0.3	5000	0.074894
0.4	5000	0.20670716549791085	0.4	5000	0.099036	0.4	5000	0.099036
0.5	5000	0.19319598716576758	0.5	5000	0.12253	0.5	5000	0.12253
0.6	5000	0.190815143813774	0.6	5000	0.137704	0.6	5000	0.137704
0.7	5000	0.16357298048141028	0.7	5000	0.123958	0.7	5000	0.123958
0.8	5000	0.07014835119096643	0.8	5000	0.048323	0.8	5000	0.048323
0.9	5000	0.14007232299946495	0.9	5000	0.178566	0.9	5000	0.178566
1	5000	0.7284075614887531	1	5000	0.749137	1	5000	0.749137
1.1	5000	0.8662812286622921	1.1	5000	0.86811	1.1	5000	0.86811
1.2	70	0.863775971704605	1.2	84	0.863761	1.2	112	0.863761
1.3	5000	0.8730608525445581	1.3	5000	0.830149	1.3	5000	0.830149
1.4	5000	0.8002733898956806	1.4	5000	0.874509	1.4	5000	0.874509
1.5	5000	0.8390365399229451	1.5	5000	0.889225	1.5	5000	0.889225
1.6	830	0.8637602698178375	1.6	5000	0.878988	1.6	5000	0.878988
1.7	5000	0.8355452194401756	1.7	5000	0.859899	1.7	5000	0.859899

Tabela 5.4 – Testy dla metody siecznych | Test wartości | Start_2 = 0.2

Dla testów z drugim punktem startowym, jako końcem przedziału przeprowadzimy testy tylko dla $\rho = 10^{-4}$.

Start	Iteracje	Wynik
0.3	5000	0.8674953731457067
0.4	5000	0.8703254783161696
0.5	1946	0.8637611708686819
0.6	5000	0.8337714572387434
0.7	15	nan
0.8	15	nan
0.9	5000	0.8686847131312921
1	191	0.009935542004618934
1.1	13	nan
1.2	11	nan
1.3	11	nan
1.4	9	nan
1.5	9	nan
1.6	9	nan
1.7	9	nan

Tabela 5.5 – Testy dla metody siecznych | Test długości kroku | Start_2 = 1.8

Start	Iteracje	Wynik
0.3	102	0.8637767977254225
0.4	5000	0.8684864573964692
0.5	5000	0.735639506192548
0.6	5000	0.8589285536732947
0.7	5000	0.8607582364902137
0.8	5000	0.8796346678148061
0.9	5000	0.7928686998713541
1	5000	0.88024314606226
1.1	3846	0.8637459262760561
1.2	32	0.8637763859645683
1.3	8	nan
1.4	12	nan
1.5	13	nan
1.6	15	nan
1.7	15	nan

Tabela 5.6 – Test dla metody siecznych | Test wartości | Start_2 = 1.8

6. Rozwiązywanie układu równań nieliniowych metodą Newtona:

Dany jest układ równań nieliniowych:

$$\begin{cases} x_1^2 + x_2^2 - x_3^2 = 1 \\ x_1 - 2x_2^3 + 2x_3^2 = -1 \\ 2x_1^2 + x_2 - 2x_3^2 = 1 \end{cases}$$

Rozwiążemy go za pomocą metody Newtona. Metoda Newtona została już przedstawiona w punkcie 2. dla równania z 1 niewiadomą. W przypadku tego układu posiadamy 3 niewiadome. Rozwiązanie będzie jednak wyglądać bardzo podobnie, z tym, że musimy policzyć nie pochodną $f'(x)$, ale Jakobian $J(X)$. Najpierw przekształcimy powyższy układ tak, aby po prawej stronie równań znajdowały się wyłącznie 0.

$$\begin{cases} x_1^2 + x_2^2 - x_3^2 - 1 = 0 \\ x_1 - 2x_2^3 + 2x_3^2 + 1 = 0 \\ 2x_1^2 + x_2 - 2x_3^2 - 1 = 0 \end{cases}$$

Wprowadźmy oznaczenia:

$$X = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

$$F(X) = \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_n \end{bmatrix} = \begin{bmatrix} x_1^2 + x_2^2 - x_3^2 - 1 \\ x_1 - 2x_2^3 + 2x_3^2 + 1 \\ 2x_1^2 + x_2 - 2x_3^2 - 1 \end{bmatrix}$$

$$J(X) = \begin{bmatrix} \frac{\partial F_1}{\partial x_1} & \cdots & \frac{\partial F_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial F_n}{\partial x_1} & \cdots & \frac{\partial F_n}{\partial x_n} \end{bmatrix}$$

Kolejne przybliżenia wektora X obliczać będziemy ze wzoru:

$$X_{k+1} = X_k - \frac{F(X_k)}{J(X_k)} = X_k - J(X_k)^{-1} F(X_k)$$

7. Warunki końca dla układu równań:

W przypadku układu równań stosować będziemy analogiczne warunki do tych, które przedstawiliśmy w punkcie 4, jednak będą one uogólnione dla X, który jest wektorem, a nie pojedynczą zmienną.

Test wartości:

Będzie przedstawiał się wzorem:

$$|F(X_k)| = \sum |f_i(X_k)| < \rho$$

W przeciwieństwie do wersji z 1 zmienną, tutaj bierzemy sumę wszystkich funkcji w wektorze F.

Test długości kroku:

Przedstawia się wzorem:

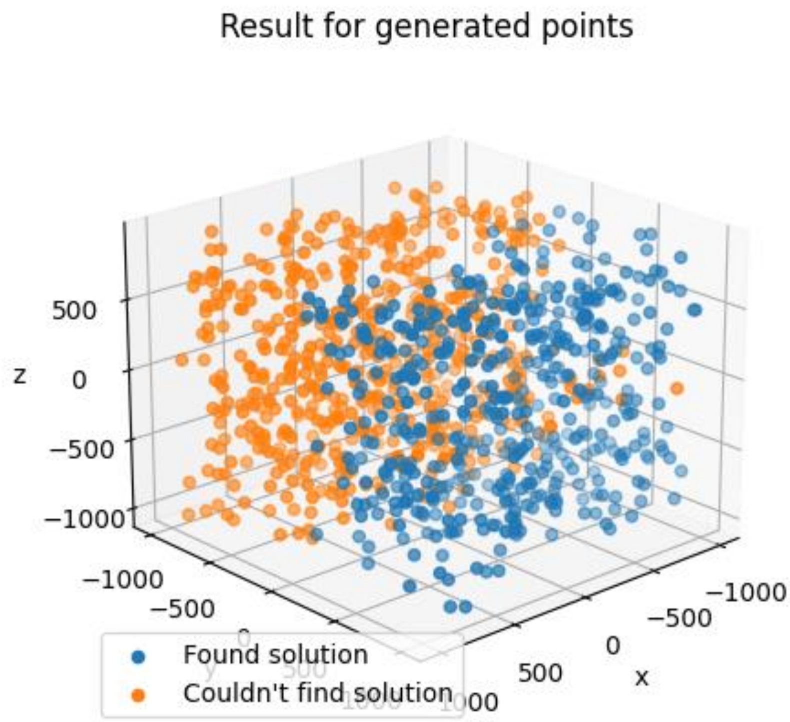
$$\sum_{k \in [0,1,\dots,n]} |x_{k+1} - x_k| < \rho$$

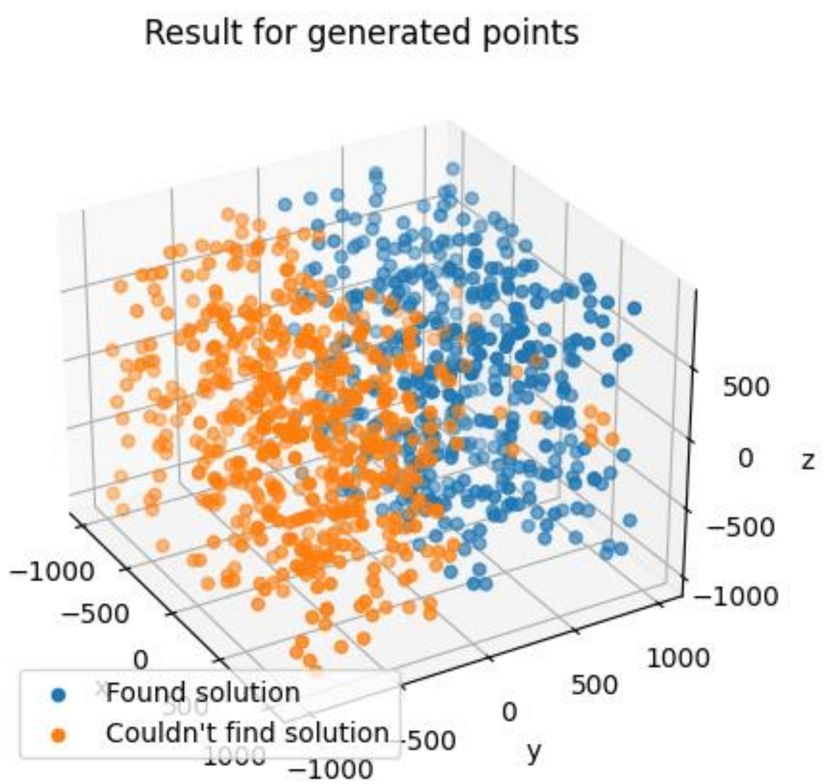
8. Testy dla zadanych punktów startowych:

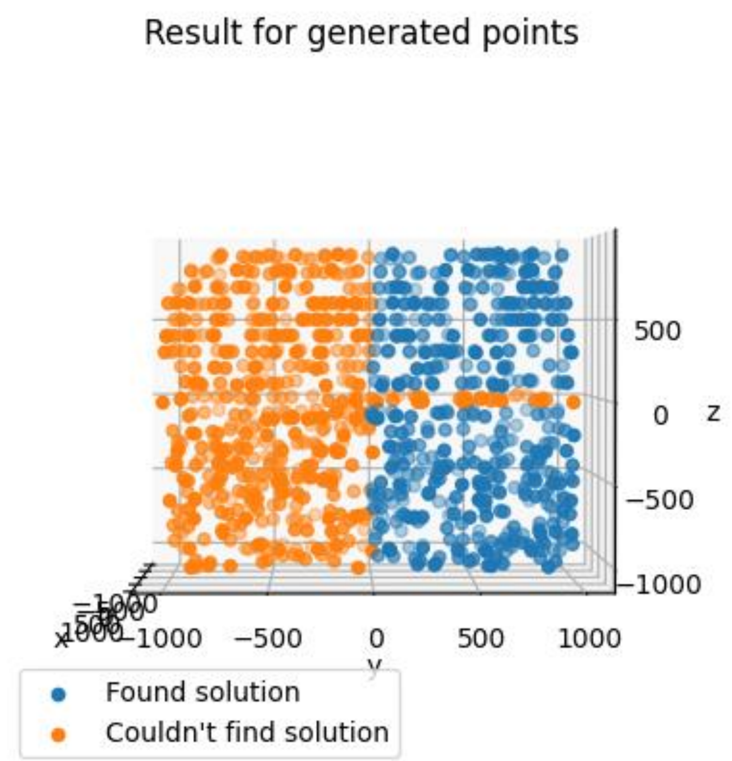
Dla przetestowania algorytmu wygenerowaliśmy 1000 losowych punktów o współrzędnych $x \in [-1000, 1000],\ y \in [-1000, 1000]\ oraz\ z \in [-1000, 1000]\ \ i\ \ z\ mod\ 100 = 0$. Współrzędna z została ograniczona w ten sposób, żeby nawet przy bardzo dużej ilości punktów wykres sukcesów i porażek był przejrzysty.

Wszystkie znalezione przez program rozwiązania (są to również wszystkie rzeczywiste rozwiązania tego układu):

$$\begin{cases} x_1 = -1 \\ x_2 = 1 \\ x_3 = -1 \end{cases} \quad \begin{cases} x_1 = 0.5 \\ x_2 = 1 \\ x_3 = 0.5 \end{cases} \quad \begin{cases} x_1 = 0.5 \\ x_2 = 1 \\ x_3 = -0.5 \end{cases} \quad \begin{cases} x_1 = -1 \\ x_2 = 1 \\ x_3 = 1 \end{cases}$$





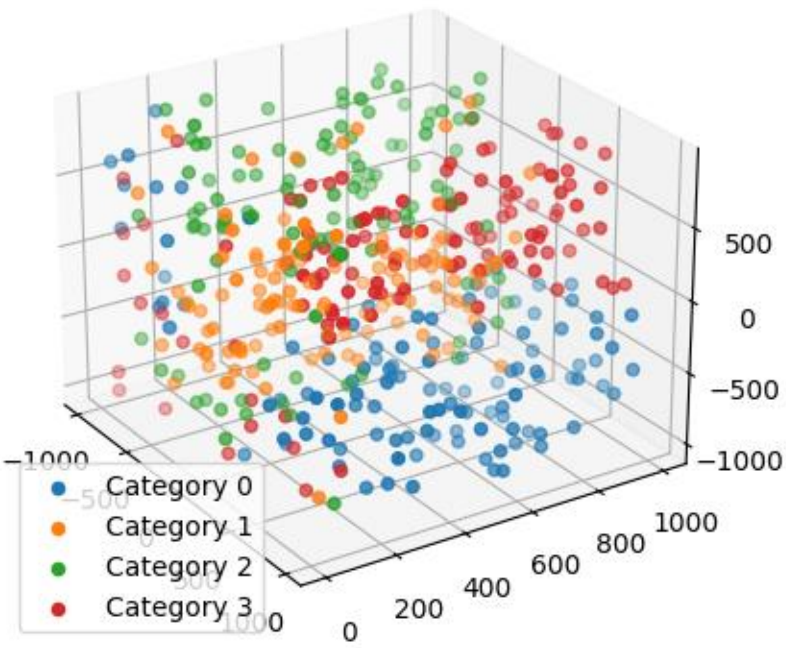


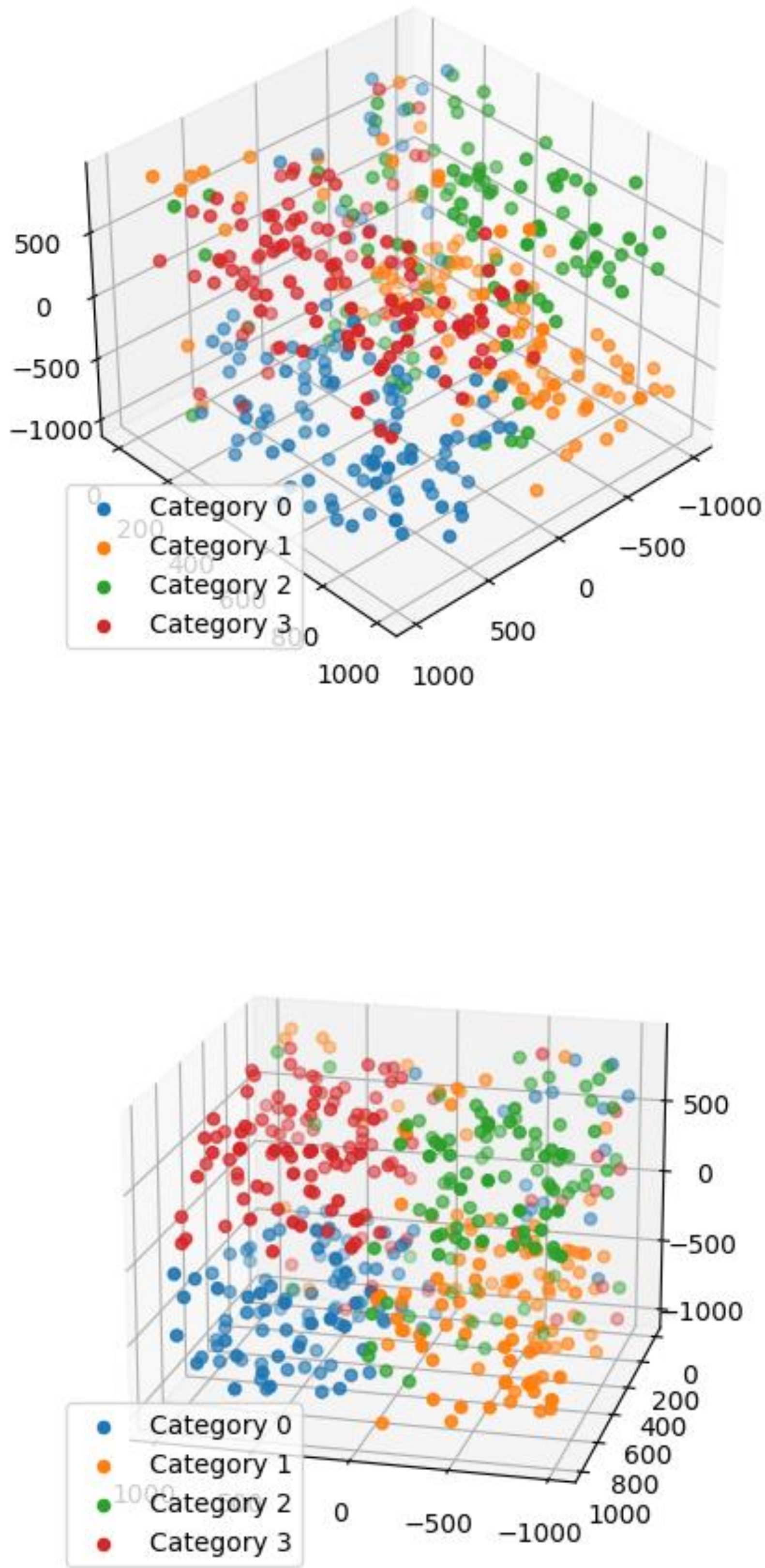
Wykresy 8.1 – Klasyfikacja testowych punktów startowych w zależności od znalezienia rozwiązania

Średnia ilość wykonanych iteracji dla punktów startowych, dla których zostałe znalezione rozwiązania: 20.3974082.

Ponieważ rozwiązanie zadanego układu posiada 4 rozwiązania, zadamy 4 kategorie punktów, z których każdej przypiszemy kolor i pokolorujemy odpowiednio punkty startowe, dla których zostało znalezione rozwiązanie, w zależności od tego które z rozwiązań zostało znalezione dla danego punktu.

Kategoria:	0	1	2	3
	$\begin{cases} x_1 = -1 \\ x_2 = 1 \\ x_3 = -1 \end{cases}$	$\begin{cases} x_1 = 0.5 \\ x_2 = 1 \\ x_3 = 0.5 \end{cases}$	$\begin{cases} x_1 = 0.5 \\ x_2 = 1 \\ x_3 = -0.5 \end{cases}$	$\begin{cases} x_1 = -1 \\ x_2 = 1 \\ x_3 = 1 \end{cases}$
	Niebieski	Pomarańczowy	Zielony	Czerwony





Wykresy 8.2 Klasyfkacja punktów względem znalezionego rozwiązania

9. Podsumowanie:

Metody zaprezentowane w sprawozdaniu pozwalają na efektywne rozwiązywanie równań i układów równań nieliniowych. Wynik jest istotnie zależny od wybranego punktu startowego. Nie dla każdego punktu startowego wynik zostanie znaleziony – istnieją „złośliwe” punkty, które dla przedstawionych metod sprawią, że wynik będzie zbiegał do nieskończoności (w tabelach z wynikami ozaczana jako **nan**).